AFRL-IF-RS-TM-1999-2
In-House Technical Memorandum
July 1999

STATISTICAL PATTERN RECOGNITION TOOL UPGRADES

Shaun P. Montana

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TM-1999-2 has been reviewed and is approved for publication.

APPROVED:

GERALD C. NETHERCOTT
Chief, Multi-Sensor Exploitation Branch
Info & Intel Exploitation Division
Information Directorate

FOR THE DIRECTOR: John V McNamara

JOHN V. MCNAMARA, Technical Advisor
Info & Intel Exploitation Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFEC, 32 Brooks Rd., Rome, NY 13441-4114. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. REPORT TYPE AND DATES COVERED
Tech Memo June 1998 - August 1998

4. TITLE AND SUBTITLE
STATISTICAL PATTERN RECOGNITION TOOL UPGRADES

5. FUNDING NUMBERS
C: N/A
PE:62702F
PR: 4594
TA: 15
WU: 2F

6. AUTHOR(S)
Shaun P. Montana

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AFRL/IFEC
32 Brooks Road
Rome NY 13441-4114

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-IF-RS-TM-1999-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/IFEC
32 Brooks Road
Rome NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TM-1999-2

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Andrew Noga, IFEC, 315-330-2270

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Further upgrades have been made to a previously developed statistical pattern recognition and analysis software tool. This tool, referred to as STATPACK, has been developed in the MATLAB processing environment. The upgrades include Fisher discriminant projection capabilities, for data structure analysis.

14. SUBJECT TERMS
coordinate vector projections, eigenvector projections, fisher's linear discriminant, statistical pattern recognition

15. NUMBER OF PAGES
36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASS	UNCLASS	UNCLASS	UL

Table of Contents

1.0 Introduction

STATPACK is a statistical pattern analysis and recognition package developed for use in the MATLAB [1] program environment. It is loosely based on the OLPARS (On-Line Pattern Analysis and Recognition System) program [2]. Initially developed as in-house work from June to August 1996, the package underwent expansion from June to August 1997 and revision in September and October 1997. Documentation is available concerning the initial effort and the first expansion, and can be found in reference [3], Rome Laboratory In-House Report RL-TM-96-8, *"A Statistical Pattern Recognition Tool"* and reference [4], Rome Research Site In-House Report RL-TM-97-4, *"Enhancements to a Pattern Recognition Tool"*. The following report documents in-house work performed this past June to August 1998, under the Rome Research Site Summer Engineering Aide Program. During this time, changes to the package added new projection tools and addressed multi-platform capability issues.

This report also covers the initial results of bit stream data analysis performed using STATPACK. File bit stream data was provided by Edward Rice through Research Associates of Syracuse Inc. (RAS), covering various file types (text, html, GIF) in both scrambled and encrypted form. Essentially, this involved extracting predominantly entropy based feature vectors from segments of the data files, and associating these vectors with the type of file from which they were extracted. Rice formatted the data for use by STATPACK, and the package was used to provide an initial analysis of the data. The author also made use of the data as a means of testing the STATPACK package's ability to deal with data files larger than the nasa.dat file used in development. Rice's data also served as an excellent performance test with regards to finding errors in the code.

Finally, this report contains an evaluation of STATPACK as it exists to date and many suggestions as to future enhancements and additions to the program. These suggestions, from the author and many other sources, should provide a glimpse as to where development should occur within STATPACK.

2.0 Continued Enhancement and Testing of STATPACK

Three specific tasks defined the course of work over the summer. These included the development of a Fisher projection and classification scheme, analysis of Rice's bit stream data, and modifying STATPACK to work on various platforms. Initial work covered getting STATPACK to run under a Windows NT 4.0 platform. (Up to this point, all coding and testing of the STATPACK package was accomplished under MATLAB v4.2c on a machine running Windows 3.11.) This task completed, initial coding of a classification scheme based on the Fisher Discriminant resulted in the frame/outline of the function FISHERC. The next logical step, was to determine the mathematics behind the Fisher Discriminant. After much research, the mathematics used by Johnson and Wichern [5] was chosen, leading to the function S2FSHP. A second Fisher Projection function, S2FSHPO, was developed based on the OLPARS function S2FSHP [2], but made use of the mathematical procedure found in Johnson and Wichern. Coding was then put aside to begin some initial analysis of Rice's data. Once incomplete vectors were identified and removed, the data was viewed using the existing one and two-dimensional structure analysis functions and the newly developed Fisher Projections. The data also served to test these functions and STATPACK's overall ability to handle files in excess of four thousand vectors. STATPACK's only complete classifier, NMCLASS, was not used to view the data. Problems resulting from FILEIN, documented below, prevented this. The task of converting to

UNIX compatible MATLAB code was interspersed with the data analysis. However, the sole available machine for testing UNIX compatibility ran MATLAB v5.2.1. Thus, the conversion task took the form of two steps, first a conversion to MATLAB v5.2.1 for Windows NT 4.0 and then to MATLAB v5.2.1 for UNIX.

2.1 STATPACK Versions

As of the writing of this report, there are five different versions of STATPACK. This is understandable, since the package is in the development stage. The intent is to ultimately revise this summer's version, incorporating all changes. Until then, it is useful to distinguish between the different versions. The original is known as "STATPACK" but will be referred to as "STATP96" to avoid confusion. "STATPACK" will refer to the package in general, not necessarily a specific version. STATP96 contains the first summer's work, documented in [3]. The second version is "STATP97", which contains the first two summer's work, documented in [3] and [4], and Floyd's enhancements following both summers, documented in [4] and [6]. The third version, "STATP98", contains the changes to the MATLAB 4.2c code written this summer (Fisher Projections, etc.) and documented in this report. The fourth identifiable version, "STATP985", is essentially the STATP98 code with changes made so it will run specifically under MATLAB v5.2.1 on a Windows 95/NT machine. It is documented, to a lesser degree, in this report. Another version that exists, "STATP97B", is a copy of the STATP97 code to which Floyd added UNIX and MATLAB v5.2.1 compatibility. It is intended to run under MATLAB v4.2c or v5.2.1, on either a UNIX or Windows 3.1x/95/NT machine. Again, it is expected that the STATP98 code will eventually incorporate these compatibilities.

2.2 STATPACK and Windows NT

For the past two summers STATPACK's development occurred using MATLAB v4.2c on a 486/66 PC running Windows 3.11. This summer, STATPACK was enhanced under MATLAB v4.2c on a Gateway 2000 P5 running Windows NT 4.0. Initial running of STATPACK resulted in errors, which was strange as the errors occurred in code that had not been modified since last summer. Investigation lead to the suspected cause of the errors: Windows NT multitasking. The "dos" command used by FILEIN to create a directory is multitasked and not executed immediately. As such, when the next line of code attempts to change to this directory, or store data there, it does not find the directory and stops the program with an error. A small function, DELAY25, was written to get around these errors. It keeps the function in a loop until the directory is created. Initially tested successfully, this fix seemed to solve the errors. Successive calls to FILEIN would sometimes result in the program almost finishing, then coming to a stop by freezing both STATPACK and MATLAB. Killing the program would be the only way of getting around this, and running FILEIN a second time would result in no freezes or errors. This was ignored and explained away as a software error, as the program froze many times in the past. This seemed to be a valid conclusion, as FILEIN never produced such errors when it was run on MATLAB v5.2.1 later in the summer. Near the conclusion of the summer, tests on the program for presentation purposes resulted in errors with NMCLASS, the nearest mean classifier function. Investigation seemed to show the delay function as the cause of the freeze. This was further implied when the NMCLASS code from STATP97 ran flawlessly on Windows 3.11 but failed to run under Windows NT 4.0 due to the multitasking problem. The STATP98 version of the NMCLASS code failed to run on either 3.11 or NT 4.0. The only difference between the two sets of code is the delay function. Alternatives were explored, but only briefly, as this was discovered with only a short amount of time left in the author's summer tour. The best solution at this time, seems to be running the appropriate version of STATPACK written in MATLAB v4.2 code only on Windows 3.1x machines, or running the appropriate version written in MATLAB v5.2.1 code Windows NT 4.0. It is unknown how the software functions on Windows 95 for either version of MATLAB, as this has never been tested.

2.3 New Two Dimensional Analysis and Classifier Functions

Two two-dimensional analysis functions were added to STATPACK. These functions, S2FSHP and S2FSHPO, are two different versions of a projection that uses the Fisher Discriminant. The first, S2FSHP, is based on an algorithm found in [5] and is chosen by selecting "Johnson and Wichern" under the "Fisher Projections" menu, which is under the "Analysis" "2D-Structure" menu. Two eigenvectors calculated from the Fisher Discriminant and selected by the user are used to project the data. The second, S2FSHPO, is based on the OLPARS [2] function S2FSHP. It determines the Fisher Discriminant based on the same mathematics as S2FSHP (the STATPACK version) but uses two pairs of classes to determine the two eigenvectors. The user determines the eigenvectors in S2FSHPO by selecting the pair of classes used to generate each eigenvector. These two eigenvectors define the plane onto which the data will be projected.

Work on another classifier for STATPACK, called FISHERC, was started before the structure analysis functions. Based on the OLPARS function FISHER, this function has the user select a number of thresholds (up to four) and calculates them for each potential pair of classes. Each vector is then multiplied by the Fisher Discriminant, and classified according to decision logic based on a comparison of the product with the thresholds. These functions are further described in proceeding sections.

2.4 STATPACK and MATLAB v5.2.1

The first step in converting to UNIX compatible code was to convert to MATLAB v5.2.1. A version of STATP97 (STATP97B) had already been coded for use under MATLAB v5.2.1, but the author was unaware of this until the writing of this report. Floyd, who was responsible for STATP97B, was also unaware the author was going to be working with MATLAB v5.2.1 until his presentation near the end of the summer. As such, STATP985 came into being as the STATP98 code converted to run under MATLAB v5.2.1. STATP985 contains the summer's additions but is incomplete. However, the only code of STATP985 not fully tested under MATLAB v5.2.1 is the set of functions available from the "Classify" menu. STATP97B is a total conversion but does not include this summer's additions. It does contain the ability, however, to run under both MATLAB v4.2c and v5.2.1. The author intends to add the changes made in STATP98 to STATP97B and compare that with STATP985, if possible.

There were a few necessary changes to be made in STATP985. The first noticeable change was the position of any full size figure window. Setting the position vector as [0,0,1,1] resulted in the title and menu bar of the window overflowing the top of the screen. Consultation with Floyd revealed that MATLAB v4.2c would have this problem as well but forced the top of the figure to match the top of the screen. However the bottom then overflowed. This can easily be tested and shown. Floyd created a function, POSFIG, to determine the screen size and set the values for the position vector based on the screen size. The function returns two values, a position vector for a figure with a menu, and a vector for one without. He then modified the

STATPACK function to globalize these variables and included POSFIG immediately after setting the main STATPACK global variables (SPROOT, SPDATA, SPNODE, DIRSEP) (see [4]). Similar additions can be made to other functions which produce full-screen windows. STATP97B is the only version of the code to currently contain this modification. Floyd also created a function to determine which version of MATLAB STATPACK was using. Called MLVER, the function's output can be used to choose between different code needed for different versions. This allows one set of STATPACK code to exist that runs on both versions, instead of having separate sets of code for the two different versions. STATP97B is the only version to currently make user of MLVER.

2.5 STATPACK and UNIX

With the appropriate changes made to STATP98 code for operation under MATLAB v5.2.1 and with the existence of STATP97B, UNIX compatibility can be tested. Floyd added UNIX compatibility code to STATP97 in September and October 1997. The only changes required to the STATP985 code were to make certain all function names in the code were in lowercase and the filenames themselves were all in lowercase. Whereas Windows 3.1x/95/NT is not case sensitive, UNIX is, and differences in case will cause errors on a UNIX machine. Also, returns at the end of lines in some files showed up as the character '^M' when the package was transferred to a UNIX machine, causing errors. As of this writing neither STATP985 nor STATP97B has been tested by the author on a UNIX machine. In theory, both should function perfectly. Testing (and error correction, if needed) is still required and should be done before the package is used on UNIX machines.

3.0 Overview of STATPACK

The batch file described in [6] allows a user to install STATPACK on a chosen directory on their machine which also has MATLAB v4.2c or higher on it. As indicated in Floyd's paper, "The batch file creates the base directory, all needed subdirectories, copies source and data files into these directories, and writes two new files: stpkroot.m and pathsp.m. Stpkroot.m declares global path variables SPROOT, SPDATA, SPNODE, and assigns the base directory name to SPROOT. Pathsp.m creates a MATLAB search path for STATPACK by pre-pending search path to the nominal MATLAB search path." It is recommended, though not required, that the user change MATLAB's startup.m file to include the pathsp.m commands. This batch file is run by typing 'a:mksp [drive]:[dir]' at the command prompt, with the disk containing the file inserted into the A drive. [dir] is the directory the user wishes to enter STATPACK into. A batch file was also created to save and backup STATPACK's code. This is run by typing 'a:busp [drive]:[dir]' at the command prompt. Here, [dir] is the directory that STATPACK can be found in.

The program itself is run by typing 'statpack' at the MATLAB command line. This begins the program and displays a box of information about the program, and how to proceed. It also contains the name of the most recently selected data node and sub node. Once a data file is loaded using FILEIN, the user can proceed to use a number of different functions accessed through STATPACK menus.

3.1 Startup Screen and Menu Changes

The STATPACK main screen has been modified to include a small white text box in the upper right hand corner. This box displays whatever the current node is and changes appropriately as nodes are added, selected, and removed. This addition, by Floyd, eliminates the need for the "Current" option under the "Node" menu. The options in this menu were relabeled to provide the user with a better description of each option. Now, the options under "Node" include "Select Node", "Show Classes and Tags", and "Remove Node and Data". These correspond to the old labels "Select", "Show", and "Remove", documented in [4]. A third option, "Fisher Projections", is available under the option "Two Dimensional" of the "Analysis" menu. It contains two options, "Johnson and Wichern", corresponding to S2FSHP, and "OLPARS", corresponding to S2FSHPO. The incomplete function FISHERC is available for selection under the "Classify" menu but will result in an error if used. Finally, the "Analysis" option under the "Help" menu was split into a "1D Analysis" and a "2D Analysis" due to the growing number of analysis functions.

3.2 Two-Dimensional Fisher Projections

The two two-dimensional analysis functions added to STATPACK supplement the already existing Feature Projection (S2CRDV) and Eigenvector Projection (S2EIGV). Each of the Fisher Projections provides the user with another way of determining outliers and seeing how classes separate. Each program also follows the same programming conventions and format as S2CRDV and S2EIGV. Both make use of the MFEATURE, PLOT2D, PICKVEC, HIDE, and IDCLICK2 functions in the same ways as the already existing functions. Both offer the same menu options with some additions particular to the way the function handles projecting the data.

3.2.1 Two-Dimensional Analysis Plot Menu Changes

Menus for S2FSHP and S2FSHPO are the same as the menu found in the S2EIGV function, documented in [3]. The only addition for S2FSHP is the "Scale/Unscale Eigenvectors" option under the "Select" menu. This option allows the user to reverse the choice made during the program's execution of scaling the eigenvectors or not. There are four different options available under the "Select" menu on S2FSHPO plots. The first two options, "Classes (First Pair)" and "Classes (Second Pair)" allow the user to go back and reselect either both pairs of classes or just the second pair of classes that are used to calculate the Fisher Discriminant. The other two options, "Scale/Unscale Eigenvectors (First Pair)" and "Scale/Unscale Eigenvectors (Second Pair)", allow the user to go back and either scale the eigenvectors or not for either both or just the second pair of classes. The "Select" option under the "Help" menu has also been modified to reflect the new options available.

3.2.2 Two-Dimensional Analysis: S2FSHP

The first two-dimensional Fisher projection is based on an algorithm and mathematics found in [5]. The first part of the program, which deals with data loading, variable initialization, and removal of selected features, is an almost carbon copy of the related section of code found in S2EIGV. The code changes after the selected features are removed, if any. Then, instead of doing a covariance calculation, S2FSHP calls the function FJ2 to calculate the Fisher Discriminant. FJ2 returns four variables: the eigenvalues and associated eigenvectors, and s and $\mathbf{S}_{\text{pooled}}$, both of which are described in Section 4 below. FJ2 gives the user the option of scaling the eigenvectors or not (as shown again, in section four below), and then the code resumes looking like S2EIGV. The list of eigenvalues corresponding to eigenvectors is displayed and the user is asked to select two of a list that is no longer than s. This is different than S2EIGV, where the list of eigenvalues is equal to the number of features not removed from the calculation. The data for plotting is coded the same way as S2EIGV and then the PLOT2D function is called to plot the data. PLOT2D was changed just to alter the labels on the axes and the title on the graph, and to return to S2FSHP when it called PLOT2D. Menu options that are unique to S2FSHP are described above.

S2FSHP provides, for the nasa.dat set, one of the best projections in term of separating the data. There is little, in some cases no, overlap to be found between similar classes and there does not appear to be many outlier vectors. Reference [5] also provided a sample data set of crude oil data that was used to general a plot. Entering this data into STATPACK, however, did not give the same results as found in the plot in [5]. The results were similar, however; the plots seemed to be identical except that one was flipped over the x-axis. The reasons for this discrepancy are unknown, especially since the numerical results of the calculations performed by FJ2 match the results of the calculations in [5] almost exactly.

3.2.3 Two-Dimensional Analysis: S2FSHPO

The second two-dimensional Fisher Projection is based on the OLPARS function called S2FSHP but makes use of the mathematics found in [5]. It too follows the same code as S2EIGV until after the data has been loaded, variables initialized, and selected features removed. After this, the user is presented with a screen that lists all the classes found in the data set, and is told to select two for use in calculating the Fisher Discriminant. Only two classes can be selected; any other number of choices results in an error message and a return to the selection screen. Once the function has determined which two classes were selected, it takes the data vectors of these classes and separates them to be sent to FJ2. FJ2 returns the same results as when used in S2FSHP, but this time, only one eigenvector is available for selection, as *s* has a value of one (the number of classes, two, minus one). This eigenvector is selected if its checkbox is not marked by the user. The program will then go through and offer the same list of classes and uses essentially a repeat of the above described code to determine the second pair of classes, and calculate the Fisher Discriminant for them. Once again, only one eigenvector is available for selection. The first is automatically the x-projection direction, and the second is the y. The scale/unscale option is presented, just as in S2FSHP. The code from here until the end is exactly the same as S2FSHP, with only a different title and different menu options as described above.

Out of curiosity, it was thought that the appropriate pairs of classes could be found that would result in the S2FSHPO plot looking exactly the same as the S2FSHP plot. However, after much experimentation and lack of success, this experimentation was abandoned. For the nasa.dat data, S2FSHPO does not produce as nice a separation as S2FSHP, but otherwise does an excellent job. A copy of each plot for the nasa.dat file can be found in Appendix B of this report.

3.4 Incomplete Classifier: FISHERC

A more complete description of the method of the classifier can be found in [2]. The classifier is not finished but could be with the addition of the mathematics found in FJ2. The ability to remove vectors from the classification has been coded but never tested. Two functions were written to count the number of votes needed and calculated the thresholds but these were also never tested. This was due to the lack of available data as a result of the lack of implemented mathematics to produce this data. Once these tasks are complete, all that remains to the classifier is to take the code from NMCLASS that places the vectors in appropriate subnodes and modify it for use for FISHERC. SHLIST also needs to be modified to show the list of vectors that are removed from the classification. Testing this classifier will probably be the most time consuming part, as opposed to the actual coding.

4.0 STATPACK Mathematics

S2FSHP and S2FSHPO use a separate function, FJ2, to calculate the Fisher Discriminant. A known parameter, s, is determined first, $s = min\ (g - 1,p)$, where g is the number of classes and p is the number of features. There are never more than s non-zero eigenvalues and eigenvectors that result from the calculation. The basic mathematical equation for this calculation is:

$$\mathbf{W}^{-1}\mathbf{B}\ .$$

Here, **B** is called the "between groups matrix" and is calculated according to the following formula:

$$\mathbf{B} = \sum_{i=1}^{g} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T\ ,$$

where $\mathbf{m}_i$ is the p x 1 mean vector of the i^{th} class, and **m**, the overall average vector, is:

$$\mathbf{m} = \frac{\sum_{i=1}^{g} n_i \mathbf{m}_i}{\sum_{i=1}^{g} n_i}$$

with n_i representing the total number of vectors in class i. The resultant **B** is a p x p matrix. Next, **W** is calculating using the formula:

$$\mathbf{W} = \left(-g + \sum_{i=1}^{g} n_i \right) \cdot \mathbf{S}_{pooled}\ ,$$

with:

$$\mathbf{S}_i = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \mathbf{m}_i)(\mathbf{x}_{ij} - \mathbf{m}_i)^T$$

and

$$\mathbf{S}_{pooled} = \frac{\sum_{i=1}^{g} (n_i - 1) \cdot \mathbf{S}_i}{-g + \sum_{i=1}^{g} n_i}$$

W is known as the "within groups matrix". It is also p x p in size. $\mathbf{x}_{ij}$ is the jth vector of the ith class and is of size p x 1. With a little math, **W** can easily be reduced to:

$$\mathbf{W} = \sum_{i=1}^{g} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \mathbf{m}_i)(\mathbf{x}_{ij} - \mathbf{m}_i)^T$$

Calculating $\mathbf{S}_{pooled}$, which is not needed if the above formula is used, is still necessary. When the Fisher Discriminant is calculated, the eigenvalues and eigenvectors are determined. A constant c_k that makes the equation below true can then scale the eigenvectors:

$$c_k \cdot \mathbf{e}_k{}^T \mathbf{S}_{pooled} \mathbf{e}_k = 1, k = 1,2,\ldots,s \ .$$

In the above, $\mathbf{e}_k$ is the kth eigenvector of size p x 1, and $\mathbf{S}_{pooled}$ is of size p x p. It should be found that the number of non-zero eigenvalues/eigenvectors is less than or equal to s as defined above.

5.0 Bit Stream Data Analysis Results

The bit stream data provided by Rice came in the form of five different types: text files, GIF files, HTML files, Encrypted files and Scrambled files. The data consisted of 15 features, each of which represented a different characteristic of the bit streams. Some of the features included the Shannon Entropy, the Bit Stream Mean, Standard Deviation, Minimum and Range, and so on. Limited analysis performed with STATPACK showed that the data could be separated and that it was possible to tell the difference between encrypted text and scrambled text. Due to the problems with running the Nearest Mean Classifier, it is unknown how the data would have subdivided in this regard. Review of STATPACK by Rice resulted in suggestions from him as to additions that could be made to the package and where development of it could go. Rice strongly suggested that the identification of a vector be more in-depth than simply a number associated to its position in the data file. He desired that "the data provided allows a point to be tracked to a data feature name, (ex. Shannon Entropy), a complete vector name rather than just the first letter, a bit stream file source (ex. AUTHOR.V22), a source description (ex. Scrambled via V.22 Standard, Encrypted Text File), and a source subgroup (ex. 2 Kbyte of the Bit Stream)". Rice desired that outliers be able to be permanently removed from the source node as opposed to temporarily removed by different plots. Finally, Rice desired a way to select parameters that effectively discriminate between certain classes and be able to mark these. As these suggestions were given near the end of the summer, they were never discussed nor implemented. The author believes that they are worth looking into and certainly very possible, but wonders if they would be useful for other types of data analysis. He believes that additions such as this, which almost specialize STATPACK for dealing with a particular type of set of data, should be added as a "toolbox" as opposed to main package functions. Rice's data was invaluable in testing the code and provided STATPACK with its first test of how quickly it could handle a data set with over four thousand vectors. STATPACK performed the tasks quickly and with the ever-increasing speed of processors and amount of memory available, STATPACK should be quite capable of handling data sets well in excess of ten thousand data vectors without a notable amount of time.

6.0 STATPACK to Date

Classifier development is the main general direction to proceed, for future enhancements to STATPACK. STATPACK is not as powerful a tool as it could be with only one working classifier. Using FJ2 for the mathematics, FISHERC could easily be finished, adding to the list of classifiers. VIEWDATA, which shows classifier information or a confusion matrix, is still specific to the nasa.dat file and must be updated to be a generic function. This is something that could also be easily done. With regards to structure analysis, STATPACK is a powerful tool and easily allows a user to determine the useful features in a data set and potential ways for different classes to be divided. Developments in this area would be useful, but classifier development deserves a higher priority.

6.1 Future Developments and Directions

Node structure and tools as an area of development seems to follow logically from more classifiers. The initial development of the Fisher classifier raised many questions. These included: Should the user have the ability to copy a node, and thus perform many different classifications on the same random test set? Should the user be able to perform the same classification many times on the same data to compare results? Should random test sets be created for specific classifiers? Should the user be able to see a list of nodes that contain sub nodes? Affirmative answers to these questions would require only simple code changes or additions and could add a lot to the package. Will the node structure change as the levels increase beyond just the main level and the first sub level? This is a harder question to answer. As the code is written currently, certain guidelines and parameters would have to be followed in the creation of lower levels of nodes. These include always creating a random data test set when you want a classification to be performed, and having to save a list of subnodes in .mat format at every node with subnodes. If the structure reached six or seven sub-levels, this would be a lot of .mat files, each of which would need to be continually accessed. The current node box on the main screen would also need to be elongated to handle a longer path.

6.2 Acknowledgements

The last three summers of work that have culminated in the development and enhancement of STATPACK would never have been possible without the guidance and direction of Dr. Andrew J. Noga. Oftentimes it was his commentary that resulted in some of the functions common to STATPACK. Richard Floyd was also an indispensable aid, in speeding up code and suggesting more efficient coding methods. His knowledge of programming, MATLAB, and STATPACK have resulted in many of the functions the package currently uses. The author also appreciated the many suggestions from other employees of the Rome Research Site and former Rome Laboratory. He is also grateful for the opportunities extended to him by participating in the Summer Engineering Aide Program.

References

[1] MATLAB^{TM} User's Guide, The Math Works Inc., August 1992.

[2] S.E. Haehn, D.A. Morris, *OLPARS User Manual*, PAR Report #82-21; *OLPARS Software Reference Manual*, PAR Report #82-20; *OLPARS Programmer and System Maintenance Manual*, PAR Report #82-15; Contract #MDA904-80-C-0780, PAR Technology Corporation, June 1982.

[3] S.P. Montana, "A Statistical Pattern Recognition Tool," RL-TM-96-8, June 1997.

[4] S.P. Montana, "Enhancements to a Pattern Recognition Tool," RL-TM-97-4, March 1998.

[5] R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, 4th Ed., Prentice Hall, Upper Saddle River, New Jersey, 1998.

[6] R.M. Floyd, "TECHNICAL NOTE: A Statistical Pattern Recognition Tool (Modifications)," October 1996.

Appendix A:

List of STATPACK Routines

The following is a list of the routines written in MATLAB v4.2c code for STATPACK:

pathsp.m stpkroot.m s1crdv.m s2crdv.m

s2eigv.m s2fshp.m^ s2fshpo.m^ fisherc.m*^

nmclass.m ha1id.m ha1menu.m ha2id.m

ha2menu.m habins.m^ habout.m hacpg.m^

hahs.m hamenu.m hanal1.m^ hanal2.m^

haplot.m haprnt.m harange.m hasel.m

haself.m^ haselfo.m^ hclass.m hfile.m

hnode.m hstpk.m filein.m fileout.m

mdnode.m spmovie.m statpack.m plot1d.m

plot2d.m subp1d.m cdnode.m closefig.m

clrglb.m clrlist.m crdtset.m current.m

delay25.m^ delnode.m dialogbx.m editplot.m

fishchen.m^ fishduda.m^ fishjohn.m^ fishschl.m^

fishvote.m^ fj2.m^ fsize.m hide.m

idclick1.m idclick2.m isup.m mfeature.m

newclass.m overlap.m pickvec.m shownode.m

shlist.m textbox.m time.m waitbar2.m

viewdata.m*

* indicates function that is not complete
^ indicates new function

The following is a list of routines converted for use by MATLAB v5.2.1 code for STATP985:

statpack.m s1crdv.m filein.m subp1d.m

clrglb.m dialogbx.m newclass.m

The following is a list of routines converted for use by MATLAB v4.2c/v5.2.1 for STATP97B:

statpack.m s1crdv.m filein.m subp1d.m

clrglb.m dialogbx.m

The following is a list of routines developed by Floyd specifically for STATP97B:

posfig.m mlver.m

Appendix B:

Sample Fisher Discriminant Projections

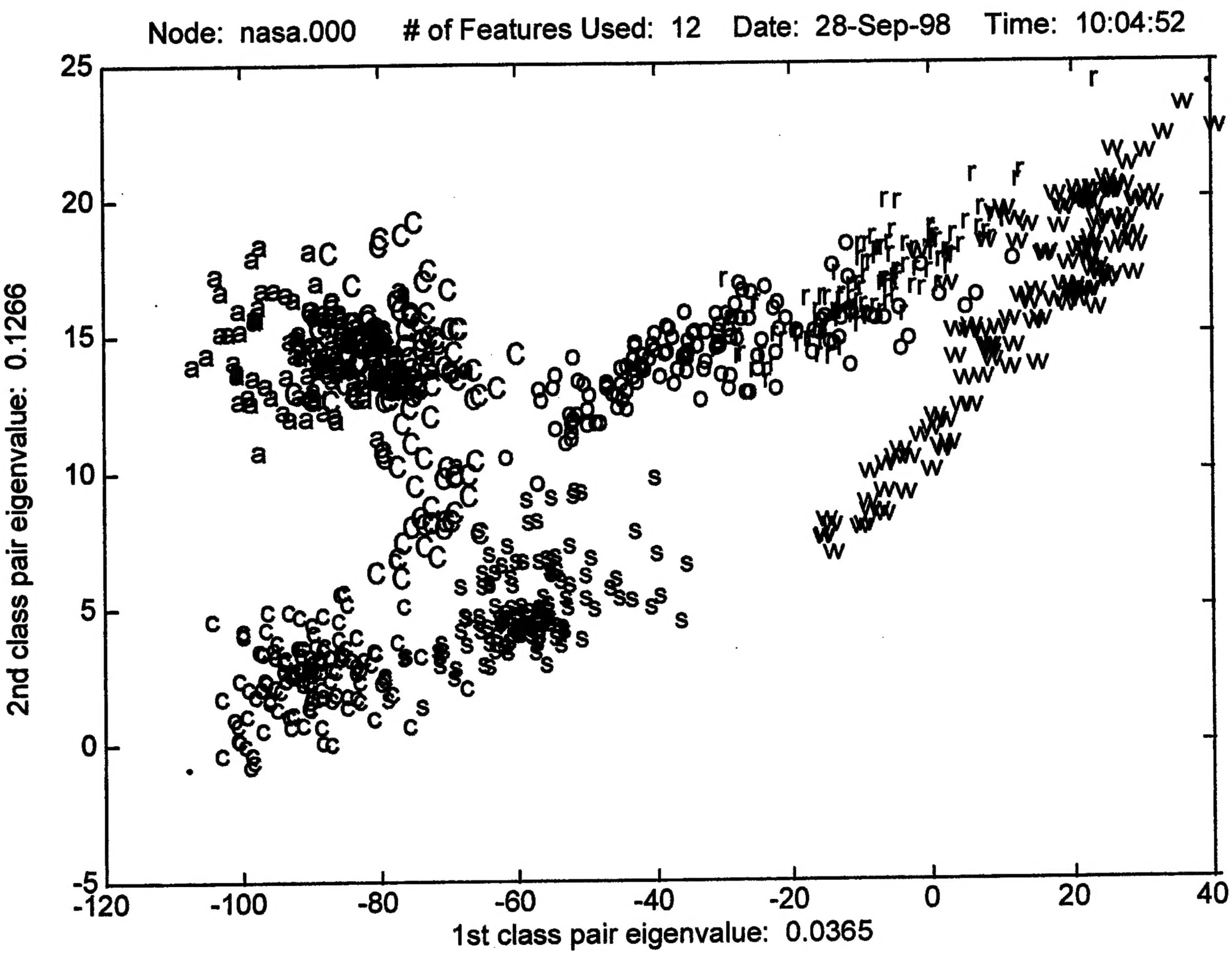

Figure B-1. Projection of nasa.dat onto Fisher linear discriminants obtained using routine S2FSHPO; classes soy and corn chosen for 1st pair, corn and oats for second pair. (All features, scaled eigenvectors.)

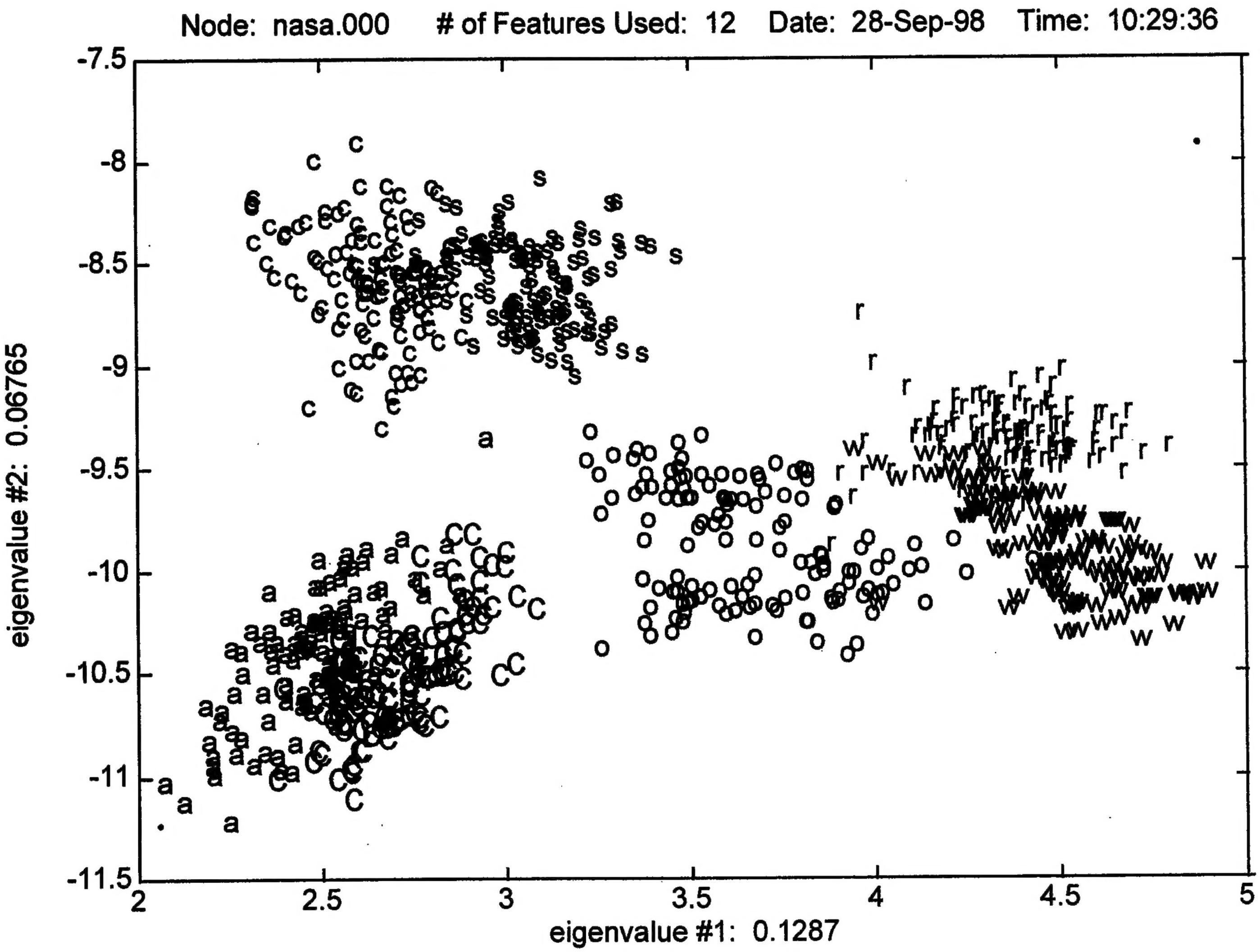

Figure B-2. Projection of nasa.dat onto Fisher linear discriminants obtained using routine S2FSHP; all features, scaled eigenvectors.